

Kernel, Licenciamento e Segurança em Ambientes *Open Source* e Proprietários

5116 - Sistemas Operativos *Open Source*

Autores:

Daniel Quaresma
Lucas Silvestre
João Correia
Vladimiro Bonaparte

Formador:

Nelson Santos



15 de maio de 2024

Conteúdo

1	Introdução	1
2	Sistemas operativos <i>open source</i> e proprietários	2
2.1	Sistemas operativos <i>open source</i>	2
2.2	Sistemas operativos proprietários	3
3	Kernel, qual a importância?	4
3.1	O que é?	4
3.2	Importância	4
4	Linux e exemplos de utilização	5
4.1	História	5
4.2	Presença em diversos ambientes	6
5	Licenciamento <i>open source</i>	7
5.1	Importância do licenciamento	7
5.2	Definição e aprovação de licenças	7
5.3	Tipos de licenças	7
5.4	Impacto do licenciamento	7
6	Segurança em sistemas operativos <i>open source</i>	8
6.1	Importância da segurança	8
6.2	Principais riscos	8
7	Software <i>open source</i> em ambientes empresários	9
7.1	Porquê?	9
7.2	Benefícios	9
8	Conclusão	10
9	Referências	11

1 | Introdução

Nos dias de hoje, a escolha entre sistemas operativos *open source* e proprietários é uma decisão crucial para as empresas. Esta escolha envolve uma série de considerações, desde o objetivo do nosso sistema ou software até às implicações legais do licenciamento e às preocupações com a segurança e estabilidade.

Neste trabalho vamos explorar estes tópicos fazendo a comparação entre sistemas operativos *open source* e proprietários e apresentando as suas características, destacando também a importância do *kernel* e as suas responsabilidades na operação de um sistema operativo.

Analisando o caso específico do GNU/Linux, vamos falar da sua história e da importância de ambos o *kernel* Linux e do projeto GNU na sua criação. Iremos ainda falar da sua presença em diferentes ambientes.

Discutimos as complexidades do licenciamento *open source*, examinando as implicações legais e práticas para empresas que optem por utilizar software de *open source*.

Investigaremos a segurança dos sistemas operativos *open source*, analisando quais os principais riscos no seu uso.

Por fim, analisaremos o processo de utilização de software *open source* em ambientes empresariais, explorando os desafios e benefícios associados à adoção deste modelo de software.



Figura 1.1: *open source* contra *closed source*

2 | Sistemas operativos *open source* e proprietários

2.1 | Sistemas operativos *open source*

Open source é um termo em inglês que significa "código aberto". Hoje, o *open source* representa não apenas um movimento tecnológico, mas também uma filosofia de trabalho que transcende a produção de software. [1]

Os sistemas operativos *open source* são desenvolvidos colaborativamente por comunidades de programadores de todo o mundo. Algumas das suas características incluem: [2]

- **Código aberto**

O código-fonte do sistema operativo é disponibilizado para o público, permitindo que qualquer pessoa estude, utilize, altere e distribua conforme os termos da licença de código aberto.

- **Transparência**

A natureza transparente do desenvolvimento de código aberto possibilita que os utilizadores examinem o código para identificar e corrigir falhas de segurança, melhorar o desempenho e adicionar novos recursos.

- **Modelo de desenvolvimento colaborativo**

O desenvolvimento de sistemas operativos *open source* é caracterizado por uma abordagem colaborativa, na qual programadores voluntários contribuem com código, correções de bugs e melhorias de forma descentralizada.

- **Flexibilidade**

Os sistemas operativos *open source* são altamente personalizáveis, permitindo que os utilizadores adaptem o software às suas necessidades específicas. Isso é particularmente vantajoso em ambientes empresariais e de pesquisa.

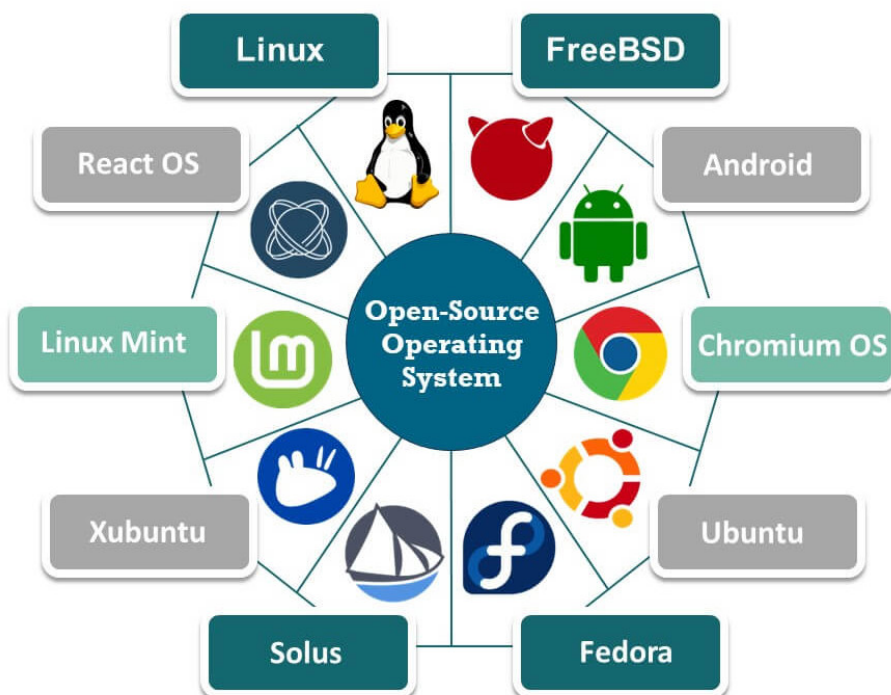


Figura 2.1: Tipos de sistemas *open source*

2.2 | Sistemas operativos proprietários

Enquanto os sistemas operativos de *open source* promovem a colaboração, a transparência e a liberdade de personalização, os sistemas operativos proprietários oferecem conveniência, suporte profissional e uma experiência de utilizador refinada. [3]

A preferência por um tipo de sistema operativo sobre o outro muitas vezes reflete valores filosóficos, necessidades específicas de uso e considerações práticas, influenciando diretamente as decisões individuais e organizacionais de adoção de tecnologia.

Os sistemas operativos proprietários são desenvolvidos e mantidos por empresas que detêm os direitos de propriedade do software. As suas características são:

■ Código Fechado

Os sistemas operativos proprietários são desenvolvidos por empresas que mantêm o código em segredo, limitando o acesso dos utilizadores à sua inspeção e modificação.

■ Controle Centralizado

As empresas que desenvolvem sistemas operativos proprietários exercem um controlo centralizado sobre o desenvolvimento, distribuição e suporte do software, o que pode limitar a flexibilidade e a capacidade de personalização pelos utilizadores.

■ Suporte Profissional

Os sistemas operativos proprietários geralmente são acompanhados por serviços de suporte profissional oferecidos pelas empresas, o que pode ser vantajoso para utilizadores e organizações que valorizam a garantia de suporte técnico especializado.

■ Restrições de Licença

Os sistemas operativos proprietários geralmente são distribuídos com licenças restritivas que limitam o uso, distribuição e modificação do software pelos utilizadores.

Essas restrições podem incluir proibições de redistribuição, limitações de uso em múltiplos dispositivos ou restrições de modificação do código-fonte.

■ Ciclos de Atualização Controlados

As atualizações de sistemas operativos proprietários são frequentemente controladas e fornecidas pelas empresas desenvolvedoras de acordo com um cronograma predeterminado.

Isso pode garantir uma maior consistência e estabilidade nas atualizações, mas também pode limitar a flexibilidade dos utilizadores em adotar novas funcionalidades ou correções de forma imediata.



Figura 2.2: Tipos de sistemas proprietários

3 | *Kernel*, qual a importância?

3.1 | O que é?

O *Kernel* é um componente fundamental de qualquer sistema operacional, sendo responsável pela gestão dos recursos de hardware e por fornecer uma interface entre o software e o hardware do computador. [4]

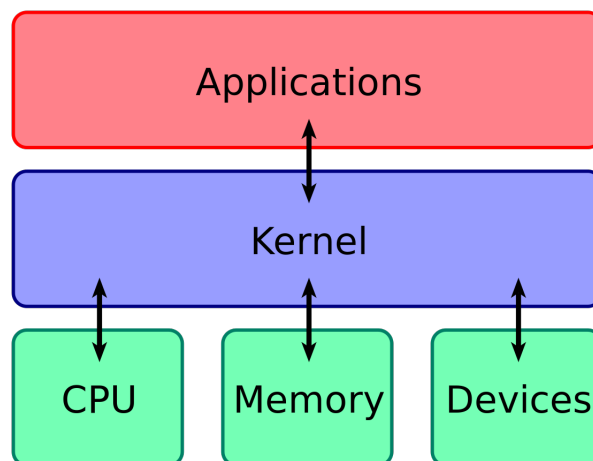


Figura 3.1: Simplificação de como opera o *Kernel*

3.2 | Importância

Algumas das responsabilidades do *Kernel* são:

- **Gestão de recursos**

O *Kernel* é responsável pela alocação e gestão dos recursos de hardware, garantindo que os processos em execução no sistema tenham acesso adequado à CPU, memória e dispositivos periféricos.

- **Abstração de hardware**

O *Kernel* fornece uma camada de abstração entre o software e o hardware, permitindo que os desenvolvedores de aplicativos escrevam código independente de plataforma. Isso facilita o desenvolvimento de software portátil e compatível com diferentes sistemas operativos.

- **Execução de tarefas do sistema**

O *Kernel* é responsável pela execução de tarefas essenciais do sistema, como o agendamento de processos, gestão de memória virtual, gestão de interrupções e controlo de dispositivos de entrada e saída.

- **Segurança e proteção**

O *Kernel* implementa mecanismos de segurança e proteção para garantir a integridade e a segurança do sistema e dos dados dos utilizadores. Controla o acesso aos recursos do sistema e impede que processos maliciosos comprometam a estabilidade do sistema.

- **Desempenho e eficiência**

Um *Kernel* eficiente é essencial para o desempenho e a eficiência do sistema operativo como um todo. Deve ser otimizado para minimizar o tempo de resposta, maximizar a utilização dos recursos de hardware e garantir uma experiência de utilizador fluída.

4 | Linux e exemplos de utilização

4.1 | História

O **GNU/Linux** é um sistema operativo *open source*, criado por **Linus Torvalds** em 1991 após a sua exposição inicial a sistemas **Unix** enquanto estudante de informática na Universidade de Helsínquia, na Finlândia.

O nome **Linux** é uma combinação do primeiro nome de seu criador, **Linus**, e do sistema operativo **Unix**, que serviu de inspiração para os seus projetos. Na altura, a maioria dos sistemas operativos era proprietária e cara, levando Linus a decidir criar um sistema operativo que estivesse disponível gratuitamente para qualquer pessoa. [5]

O Projeto **GNU**, liderado por **Richard Stallman** desde 1983, é fundamental para a história do Linux. Ao oferecer um conjunto completo de ferramentas e utilitários de software livre, o **GNU** forneceu a base essencial para várias distribuições Linux. Além disso, o **GNU** também contribuiu para a nomenclatura oficial, dando origem ao nome **GNU/Linux**. Essa colaboração entre o **GNU** e o kernel **Linux** resultou em sistemas operativos mais robustos e na capacidade de manter o GNU/Linux completamente livre. [6]

As primeiras versões do GNU/Linux eram principalmente utilizadas por entusiastas da tecnologia e desenvolvedores de software. Com o passar do tempo, a sua popularidade cresceu rapidamente, levando à sua adoção em uma ampla gama de ambientes.

O GNU/Linux é reconhecido como um dos sistemas operativos mais estáveis, seguros e confiáveis, sendo amplamente adotado em **servidores**, **supercomputadores** e **ambientes empresariais**. Com o crescimento de sua popularidade e o contínuo desenvolvimento pela comunidade, hoje, algumas das principais distribuições GNU/Linux incluem o **Ubuntu**, **Fedora**, **Arch**, **Red Hat**, **Debian**, **Mint** e **Manjaro**.



Figura 4.1: Tipos de distribuições ou "distros" de linux

4.2 | Presença em diversos ambientes

O GNU/Linux está presente em uma ampla variedade de ambientes, incluindo: [7]

■ Servidores e data centers

O GNU/Linux é amplamente reconhecido pelo seu domínio no mercado de servidores e centros de dados, destacando-se pela sua estabilidade e confiabilidade. É frequentemente utilizado para operar **redes de dados** e **data centers**.

Muitos dos equipamentos constituintes dos servidores e data centers, tais como os **routers**, funcionam com versões personalizadas e simplificadas do sistema operativo GNU/Linux.

■ Supercomputadores

A capacidade do GNU/Linux para escalar eficientemente até milhares de núcleos de processamento, aliada à sua flexibilidade para otimização em tarefas de alto desempenho, são características essenciais para o seu uso em supercomputadores.

Na verdade, o GNU/Linux é o sistema operativo preferido para a maioria dos supercomputadores, evidenciando a sua eficiência em ambientes de computação intensiva.

■ Dispositivos IoT

No âmbito da **Internet das Coisas (IoT)**, o GNU/Linux destaca-se devido ao seu tamanho compacto e à sua capacidade de adaptação para se adequar a hardware específico. Desde **eletrodomésticos inteligentes** até sistemas avançados de **controle industrial** e **veículos autônomos**, o GNU/Linux serve como uma base fiável para uma variedade de dispositivos inovadores.

■ Desktops e uso diário

Embora seja menos popular que o **Windows** ou o **MacOS** em desktops, o GNU/Linux tem observado um aumento constante na sua aceitação por parte dos utilizadores. Esta tendência deve-se à sua crescente biblioteca de programas de software baseados em GNU/Linux e a um foco contínuo na expansão da oferta de interfaces de utilizador mais amigáveis para desktop. Além disso, o sistema operativo GNU/Linux serve como a base de outros sistemas operativos amplamente utilizados no nosso quotidiano, como o **Android** e o **Chrome OS**.

■ Educação e governo

O GNU/Linux é altamente valorizado por instituições educativas e governamentais devido ao seu baixo custo e à sua capacidade de personalização. Em todo o mundo, vários governos têm implementado extensivamente o GNU/Linux nas operações governamentais e nos sistemas educacionais, aproveitando essas vantagens.



Figura 4.2: Supercomputador da IBM a correr Linux

5 | Licenciamento *open source*

5.1 | Importância do licenciamento

O licenciamento *open source* é um aspecto crucial no desenvolvimento de software, servindo como uma fundação para a inovação e colaboração tecnológica. As licenças *open source* são rigorosamente aprovadas pela **Open Source Initiative(OSI)** e asseguram que qualquer software sob estas licenças possa ser livremente utilizado, modificado e redistribuído. [8]

Este sistema de licenciamento é vital para manter a integridade e a filosofia da partilha e colaboração que são centrais para a comunidade *open source*.

5.2 | Definição e aprovação de licenças

Segundo a **OSI**, uma licença só é considerada *open source* se cumprir com a **Open Source Definition(OSD)**. Esta definição inclui uma série de critérios projetados para proteger a liberdade do utilizador e fomentar a inovação. Por exemplo, uma licença *open source* deve permitir redistribuições livres do software, acesso ao código-fonte e criação de obras derivadas.

O processo de aprovação de uma licença pela **OSI** é um pilar essencial para garantir que estas normas sejam mantidas. Através de um processo de revisão pública, a comunidade *open source* pode dar opiniões sobre novas licenças propostas para garantir que elas estejam alinhadas com os padrões estabelecidos.

Este processo não apenas protege os direitos dos utilizadores e desenvolvedores, mas também mantém um padrão uniforme que facilita a colaboração e a partilha de tecnologia entre projetos e organizações.

5.3 | Tipos de licenças

Existem dois tipos principais de licenças *open source*: **copyleft** e **permissivas**. As licenças **copyleft**, como a **GNU General Public License**, exigem que quaisquer versões modificadas do software também sejam distribuídas com a mesma licença *open source*. Isso garante que as liberdades concedidas pela licença original sejam mantidas em todas as versões derivadas do software.

Por outro lado, as licenças **permissivas**, como a licença **MIT** e a licença **BSD**, são menos restritivas, permitindo que o software seja integrado em projetos proprietários. Essas licenças ainda garantem liberdades fundamentais, mas não exigem que as obras derivadas sejam distribuídas sob os mesmos termos *open source*.

5.4 | Impacto do licenciamento

O impacto do licenciamento *open source* é profundo e abrangente. Este, permite que empresas, desde *startups* até grandes organizações, inovem e construam sobre o trabalho existente sem as restrições de licenças de software proprietário.

Este ambiente de inovação aberta tem levado ao desenvolvimento de tecnologias significativas em campos como servidores **web**, **smartphones**, **automação empresarial**, **computação em cloud** e a **economia partilhada**. O licenciamento *open source* apoia a inovação contínua e a disseminação rápida de tecnologias emergentes, beneficiando tanto os desenvolvedores individuais quanto a indústria tecnológica em larga escala.

6 | Segurança em sistemas operativos *open source*

6.1 | Importância da segurança

O software de *open source* tornou-se amplamente utilizado nos últimos anos devido à sua natureza colaborativa e pública, o que o torna conveniente tanto para os desenvolvedores como para os atores maliciosos.

Quando adversários descobrem que uma aplicação está exposta a uma vulnerabilidade conhecida publicamente, podem atacar qualquer aplicação desenvolvida utilizando esse código de *open source*. Casos como as vulnerabilidades do **Log4j** e do **Apache Struts** demonstram que isso representa um risco real e, por vezes, grave para as organizações. [9]

6.2 | Principais riscos

A maioria das aplicações nativas em *cloud* depende de componentes de *open source*. Contudo, devido à ausência de responsabilidade pela sua manutenção ou segurança, o software de *open source* apresenta diversos riscos, tais como:

■ Vulnerabilidades em dependências

Estas vulnerabilidades podem ser tanto conhecidas quanto desconhecidas. As conhecidas incluem aquelas que receberam um número de identificação de vulnerabilidade comum (**CVE**), aquelas divulgadas na Internet, aquelas presentes em bases de dados públicas de vulnerabilidades, e aquelas dentro de bases de dados privadas. Em geral, quanto mais conhecida for uma vulnerabilidade, mais urgente é a necessidade de solucioná-la.

Além de rastrear vulnerabilidades, é essencial acompanhar todas as dependências de *open source* dentro de uma aplicação. As dependências transitivas, onde uma dependência depende de outras, são especialmente preocupantes, pois são menos visíveis para ferramentas de segurança e auditorias. Portanto, é útil utilizar ferramentas ou processos que possam identificar e auditar todas as dependências em uma aplicação.

■ Riscos de conformidade com licenças

Os desenvolvedores precisam compreender cada tipo de licença de software nos pacotes de *open source* que utilizam, para poderem empregar o código de forma compatível. Isso requer conhecimento das estipulações de licenciamento e sua aplicação ao longo dos projetos.

Para garantir o cumprimento das licenças de *open source*, as organizações precisam ter uma visão aprofundada de como os componentes de *open source* estão a ser utilizados. Também é importante monitorizar continuamente as licenças, pois o proprietário dos direitos autorais pode alterar a licença de uma biblioteca.

■ Pacotes não mantidos

Os pacotes de *open source* são geralmente mantidos por um único desenvolvedor ou por uma pequena equipa, quando são mantidos. Os desenvolvedores de projetos de *open source* da comunidade não têm obrigação de manter o software, e ele é disponibilizado "como está".

Cabe aos utilizadores dedicar tempo e recursos para garantir que o código seja seguro. Felizmente, existem ferramentas úteis que podem simplificar este processo ao analisar pacotes de acordo com o nível de manutenção, envolvimento da comunidade, postura de segurança e popularidade, auxiliando na avaliação da saúde dos pacotes de *open source* utilizados.

7 | Software *open source* em ambientes empresários

7.1 | Porquê?

Software de *open source* ou proprietário? As empresas enfrentam esta questão fundamental quando se trata de escolher aplicações críticas para o seu negócio.

A decisão sobre o software adequado tem implicações a longo prazo e, portanto, deve ser considerada cuidadosamente. As empresas continuam a aumentar o seu uso de software de *open source* nos últimos anos. Mesmo que muitas empresas não estejam cientes, o software de *open source* já está presente em mais de 90% das empresas em apoio à sua infraestrutura de TI. [10]

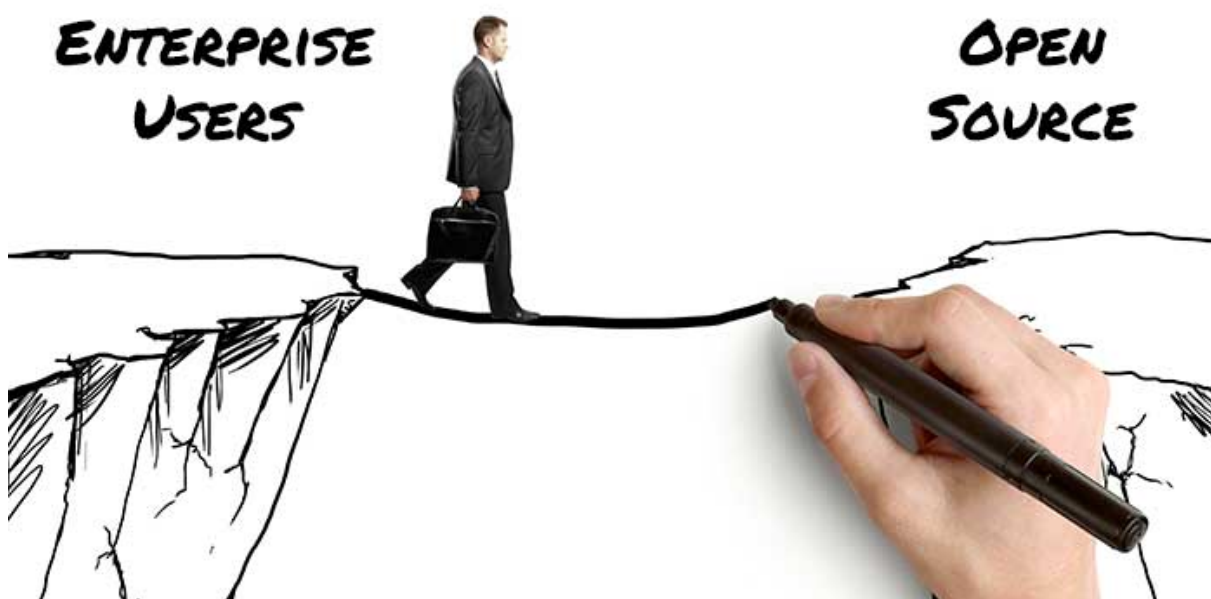


Figura 7.1: Empresas cada vez aderem mais a *open source*

7.2 | Benefícios

Algumas das principais razões que as empresas dão para a utilização de software de *open source*: [11]

- **Familiaridade** A compreensão por parte dos desenvolvedores da colaboração com comunidades, contribuições para projetos, entendimento de licenças e gestão de dependências.
- **Apoio a comunidades** O sucesso do software *open source* depende de comunidades saudáveis. Empresas podem contribuir com recursos, financiamento e feedback, fortalecendo tanto os projetos quanto a comunidade de *open source*.
- **Capacidade de influenciar o desenvolvimento dos recursos necessários** Software *open source* oferece transparência e adaptação às necessidades específicas. As empresas podem influenciar ativamente o desenvolvimento de recursos, seja contribuindo com código ou fornecendo feedback valioso.
- **Eficiência ao lidar com desafios técnicos** Acesso a uma vasta comunidade de especialistas permite solucionar desafios técnicos de forma rápida e eficaz. Fóruns online e grupos de discussão facilitam a obtenção de suporte.

8 | Conclusão

Uma análise detalhada dos sistemas operativos *open source* e proprietários revela uma série de considerações importantes para empresas que queiram adotá-los.

A importância do *kernel* como o núcleo de qualquer sistema operativo é inegável, influenciando diretamente o desempenho, a estabilidade e a segurança do sistema. O *kernel* Linux, em particular, destaca-se como um exemplo proeminente, oferecendo uma vasta gama de utilizações em diferentes contextos.

O licenciamento *open source* apresenta tanto oportunidades quanto desafios, oferecendo liberdade e flexibilidade aos utilizadores, mas exigindo uma compreensão clara das obrigações legais associadas. No que diz respeito à segurança, embora os sistemas operativos *open source* geralmente desfrutem de uma reputação de segurança robusta, é crucial adotar práticas de segurança adequadas e estar ciente das vulnerabilidades potenciais.

Para as empresas, a adoção de software *open source* em ambientes empresariais pode oferecer uma série de vantagens, incluindo custos reduzidos, maior flexibilidade e acesso a uma vasta comunidade de desenvolvedores em troca de menos suporte profissional e estabilidade.

Em última análise, a escolha entre sistemas operativos *open source* e proprietários é uma decisão estratégica que deve ser cuidadosamente ponderada, levando em consideração as necessidades específicas e os objetivos de negócios de cada empresa.

Ao compreender os diferentes aspectos envolvidos e considerando as implicações a longo prazo, as empresas devem fazer um estudo prévio, fazer um balanço entre os pontos positivos e negativos e tomar uma decisão informada sobre o tipo de sistemas a usar.

Todos os elementos do grupo participaram e alteraram o trabalho inteiro, mas o levantamento de informação de cada capítulo foi feito da seguinte maneira:

- **Daniel Quaresma:** Introdução e conclusão.
- **Vladimiro Bonaparte:** Capítulo 2 e 3.
- **Lucas Silvestre:** Capítulo 4 e 5.
- **João Correia:** Capítulo 6 e 7.

9 | Referências

- [1] Red Hat. O que é open source? <https://www.redhat.com/pt-br/topics/open-source/what-is-open-source>.
- [2] ekipa. As vantagens do open source. <https://www.ekipa.pt/as-vantagens-do-open-source/>.
- [3] Solutions Hub. Proprietary software definitions. <https://solutionshub.epam.com/blog/post/proprietary-software-definition-examples>.
- [4] Significados. O que é o kernel de um sistema operacional. <https://www.significados.com.br/kernel/>.
- [5] GeeksForGeeks. Linux history. <https://www.geeksforgeeks.org/linux-history/>.
- [6] Curious Minds. The history of free software and open source. <https://www.cmpod.net/all-transcripts/history-open-source-free-software-text/>.
- [7] Lenovo. What is linux used for. <https://www.lenovo.com/za/en/faqs/operating-systems/what-is-linux-used-for>.
- [8] Open Source Initiative. The open source definition. <https://opensource.org/osd>.
- [9] snyk. Open source security explained. <https://snyk.io/series/open-source-security/>.
- [10] pimcore. Why open source? <https://pimcore.com/en/why-open-source>.
- [11] Red Hat. The state of enterprise open source, 2022. <https://www.redhat.com/en/enterprise-open-source-report/2022>.